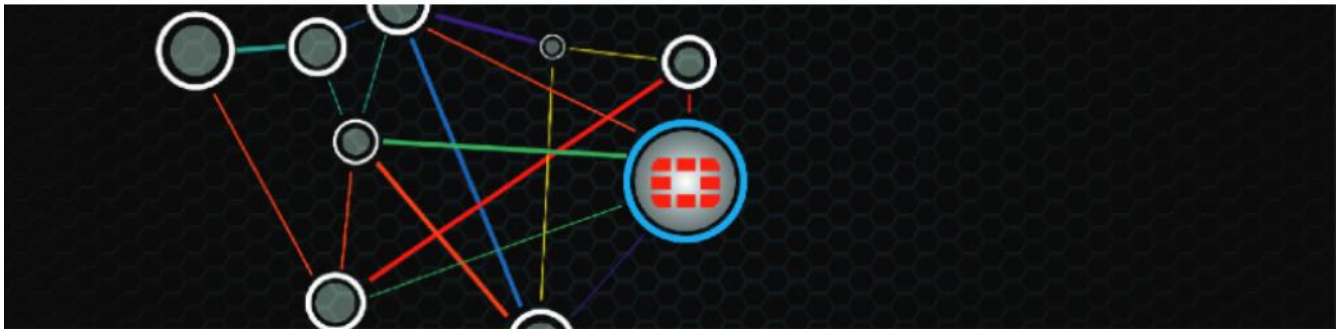
Configuration Guide



Avoiding IP Fragmentation in GRE Tunnel Deployments

Configuration Guide

Author: Donovan Williams
Consulting Security Engineer

Contents

Change Log

Revision	Date	Change Description	Owner
1	2014-06-10	Initial Release	Donovan Williams

Introduction

The purpose of this document is to explain how to avoid IP Fragmentation with the FortiGate TCP Maximum Segment Size feature when deploying FortiGate firewalls in GRE Tunnel mode.

Network Components

The following products were used:

- FortiGate 3600C FG3K6C-5.00-FW-build271
- FortiGate 1000C FGT1KC-4.00-FW-build672
- IXIA Breaking Point version 3.1 emulating clients and servers

IP Fragmentation and Reassembly Overview

IP (Internet Protocol) is used over a wide variety of transmission links. While the maximum length of an IP datagram is 64K Bytes, most transmission links enforce a smaller maximum packet length to accommodate the transmission link. This is called the Path MTU (Maximum Transmission Unit). IP allows network devices such as routers and firewalls to fragment packets in order to accommodate the respective MTU differences.
The receiving host is responsible for reassembling any fragments back to the original IP datagram. IP fragmentation involves breaking a datagram into a number of pieces that can be reassembled later. The IP source, destination, identification, total length, and fragment offset fields, along with the "more fragments" and "don't fragment" flags in the IP header, are used for IP fragmentation and reassembly. For more
Information on IP fragmentation and reassembly, please see RFC 791.

TCP Maximum Segment Size (MSS) Overview

The TCP Maximum Segment Size (MSS) defines the maximum amount of data that a host is willing to accept in a single TCP/IP datagram. This TCP/IP datagram may be fragmented at the IP layer. The MSS value is sent as a TCP header option only in TCP SYN segments. Each host of a TCP connection reports its MSS value to the each other. The sending host is required to limit the size of data in a single TCP segment to a value less than or equal to the MSS reported by the receiving host.

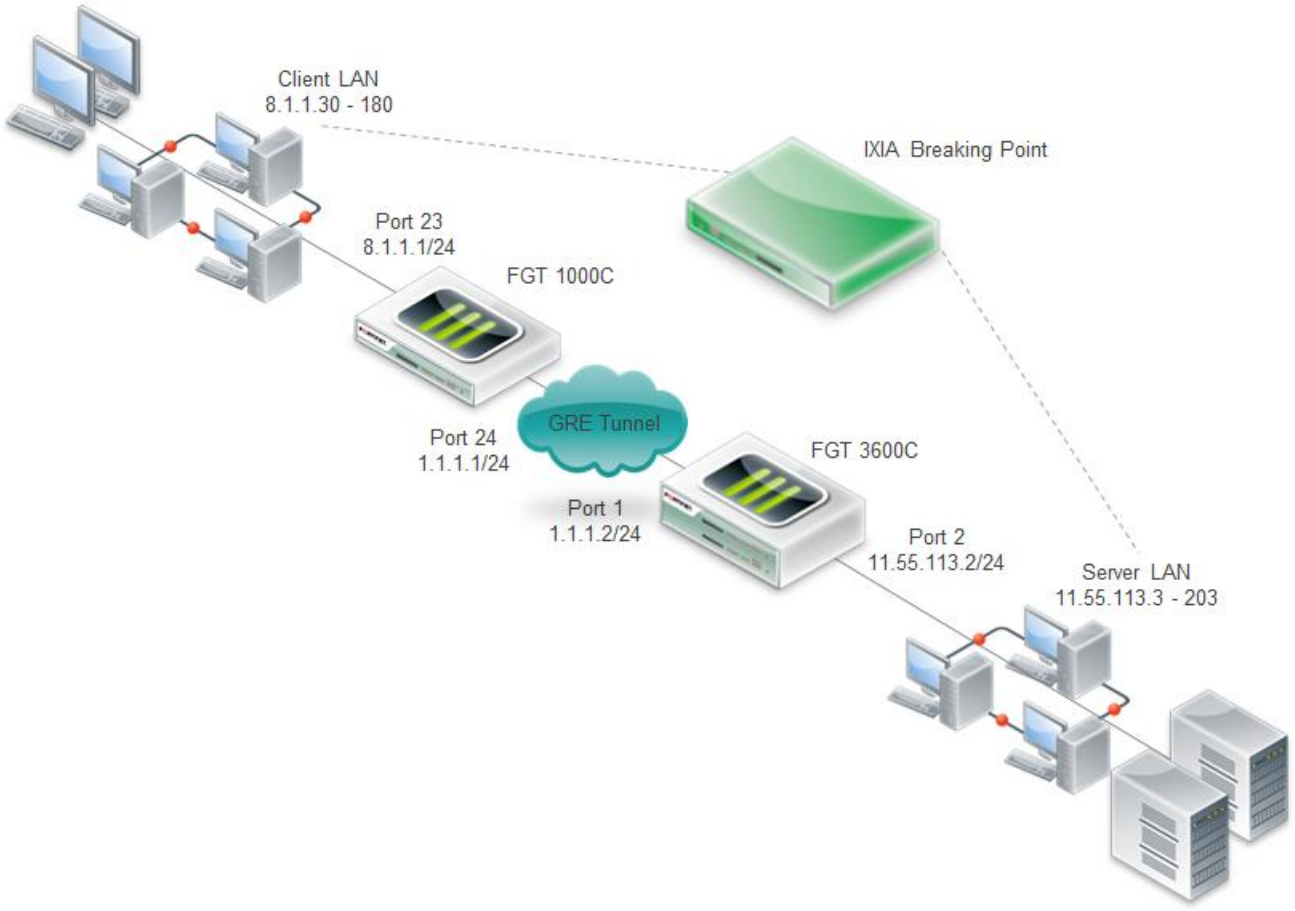
GRE (Generic Route Encapsulation) Overview

Generic Routing Encapsulation (GRE), defined by RFC 2784, is an IP packet encapsulation protocol. A GRE tunnel is used when IP packets need to be sent from one network to another, without being parsed or treated like IP packets by any intervening network devices such as routers or firewalls. GRE encapsulates packets within IP packets and redirects them to an intermediate host (router / firewall), where they are de-encapsulated and routed to their final destination.

Network Architecture

The architecture consists of 150 clients connected to port 23 on the FortiGate 1000C. The clients are part of subnet 8.1.1.0. A GRE tunnel has been implemented between port 24 on the FortiGate 1000C and port 1 of the FortiGate 3600C. The Servers are connected to port 2 of the FortiGate 3600C and are on subnet 11.55.113.0. Each client is running an HTTP 1.1 browser and downloading a 32K file from the servers. An Ixia Breaking Point was used to emulate the clients connecting to servers.

Figure 1 – Network Diagram



FGT-1000C Configuration

```
#config-version=FGT1KC-4.00-FW-build672-130904:opmode=0:vdom=0:user=admin
#global_vdom=1
config system global
    set hostname "LAB16-FG1000C-01"
end
config system interface
    edit "mgmt1"
        set vdom "root"
        set ip 10.25.16.1 255.255.0.0
        set allowaccess ping https ssh fgfm
    next
    edit "port23"
        set vdom "root"
        set ip 8.1.1.1 255.255.255.0
        set allowaccess ping
        set alias "inside "
    next
    edit "port24"
        set vdom "root"
        set ip 1.1.1.1 255.255.255.0
        set alias "to3600P1"
    next
    edit "to3600C"
        set vdom "root"
        set type tunnel
        set interface "port24"
    next
end

config firewall policy
    edit 1
        set srcintf "port23"
        set dstintf "to3600C"
            set srcaddr "all"
            set dstaddr "all"
        set action accept
        set schedule "always"
            set service "ANY"

    next
    edit 2
        set srcintf "to3600C"
        set dstintf "port23"
            set srcaddr "all"
            set dstaddr "all"
        set action accept
        set schedule "always"
            set service "ANY"
    next
end
config router static
    edit 1
        set device "to3600C"
        set dst 11.55.113.0 255.255.255.0
    next
end
```

FGT-3600C Configuration

```
#config-version=FG3K6C-5.00-FW-build271-140124:opmode=0:vdom=0:user=admin
#global_vdom=1
config system global
    set hostname "LAB01-FG3600C-01"
end
config system interface
    edit "port1"
        set vdom "root"
        set ip 1.1.1.2 255.255.255.0
        set allowaccess ping fgfm
        next
    edit "port2"
        set vdom "root"
        set ip 11.55.113.2 255.255.255.0
        set allowaccess ping
        set alias "Outside Servers "
        next
    edit "mgmt"
        set vdom "root"
        set ip 10.55.100.111 255.255.252.0
        set allowaccess ping https fgfm
        set dedicated-to management
        next
    edit "to1000C"
        set vdom "root"
        set type tunnel
        set snmp-index 40
        set interface "port1"
    next
end
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "to1000C"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "to1000C"
        set dstintf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
config router static
    edit 1
        set device "to1000C"
        set dst 8.1.1.0 255.255.255.0
    next
end
```

For detailed information on configuring GRE tunnels with static routes, please refer to the Fortinet Knowledge Base Technical Note:

http://kb.fortinet.com/kb/microsites/search.do?cmd=displayKC&docType=kc&externalId=FD31182

Fortinet TCP-MSS-Sender Option

In the diagram the clients and servers receive an MTU from their connected Ethernet interface and then calculate the MSS value (1500-40 = **1460**).
The MTU of Ethernet is 1500. The MSS number is 40 bytes smaller than the MTU because the MSS value is the TCP data size. The 20 byte IP header and 20 byte TCP header are subtracted leaving the value 1460 as the value the clients and servers send to each other as the negotiated MSS. GRE encapsulation adds an additional 24 bytes to the original IP packet (4 byte GRE header + 20 byte IP header). The clients and servers are not aware of the GRE tunnel in the path and as a result, data communications will fail due to fragmentation. The clients and servers should calculate an MSS value of 1436, (1500 – 40 – 24) to accommodate the MTU of the GRE tunnel.
The following option needs to be added to the firewall policies to set the MSS value to 1436.

tcp-mss-sender <maximumsize_int>	Enter a TCP Maximum Sending Size number for the sender. When a FortiGate unit is configured to use PPPoE to connect to an ISP, certain web sites may not be accessible to users. This occurs because a PPPoE frame takes an extra 8 bytes off the standard Ethernet MTU of 1500. When the server sends the large packet with DF bit set to 1, the ADSL provider's router either does not send an "ICMP fragmentation needed" packet or the packet is dropped along the path to the web server. In either case, the web server never knows fragmentation is required to reach the client. In this case, configure the `tcp-mss-sender` option to enable access to all web sites. For more information, see the article Cannot view some web sites when using PPPoE on the Fortinet Knowledge Center.	0

Updated Firewall Policies on the 1000C and 3600C

Fortigate 1000C Firewall Policy

```
config firewall policy
    edit 1
        set srcintf "port23"
        set dstintf "to3600C"
            set srcaddr "all"
            set dstaddr "all"
        set action accept
        set schedule "always"
            set service "ANY"
        set tcp-mss-sender 1436
    next
    edit 2
        set srcintf "to3600C"
        set dstintf "port23"
            set srcaddr "all"
            set dstaddr "all"
        set action accept
        set schedule "always"
            set service "ANY"
    next
end
```

Fortigate 3600C Firewall Policy

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "to1000C"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set tcp-mss-sender 1436
    next
    edit 2
        set srcintf "to1000C"
        set dstintf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

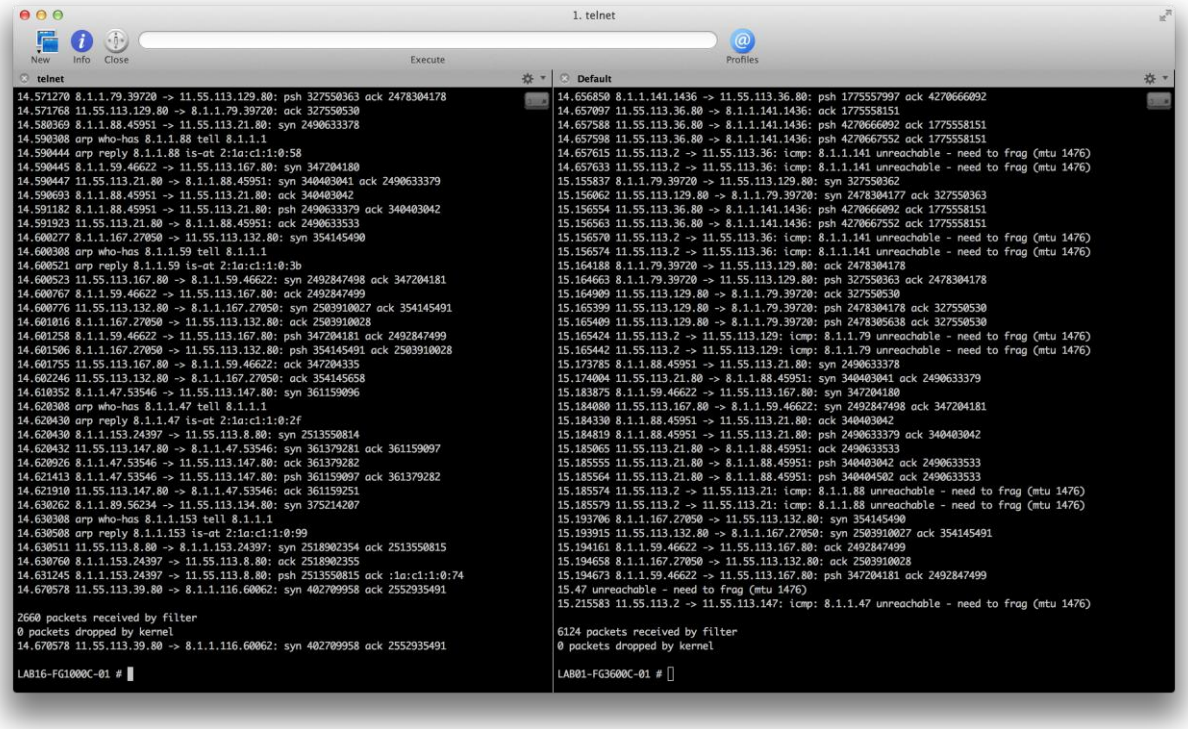
BreakingPoint Testing (Clients connecting to servers and downloading 32K files)

In this scenario, two tests were run:

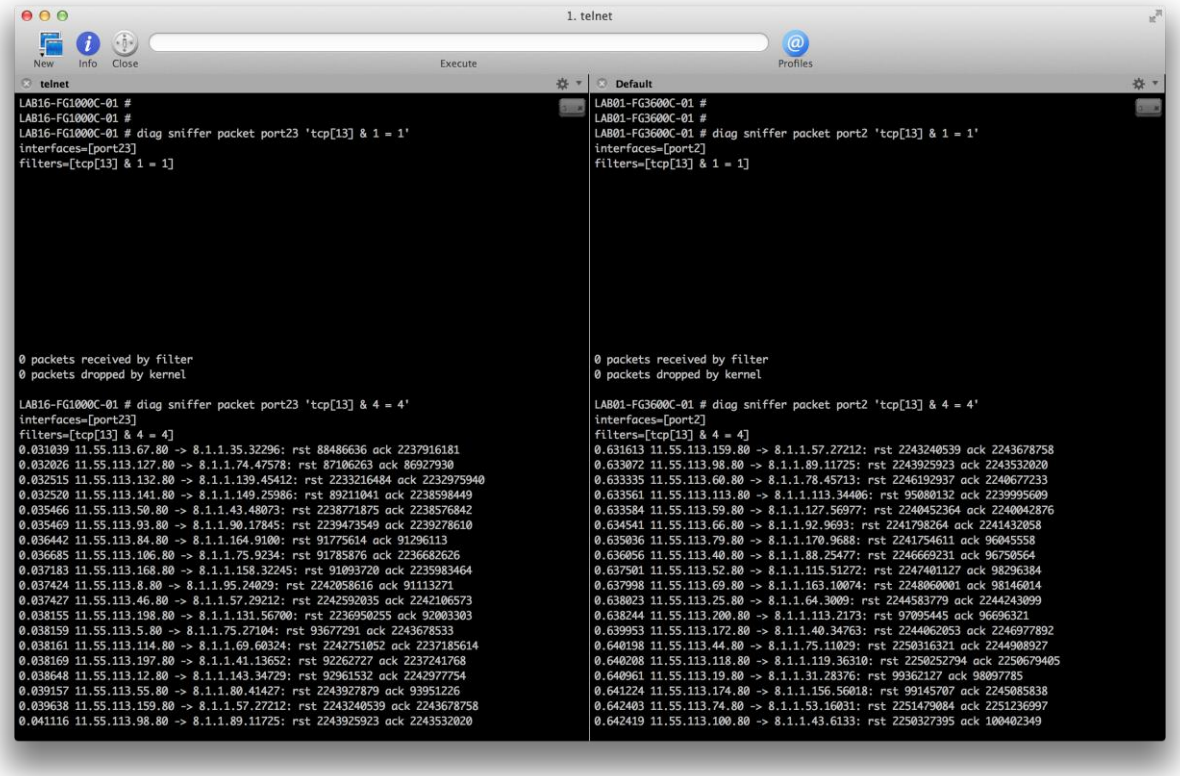
- The first test consists of clients connecting to servers and the tcp-mss-sender values are NOT configured.
- The second test repeats the first, but with the corrected tcp-mss-sender values configured.

First Test

While the clients are connecting and trying to download the files, a “diagnose debug packet” was implemented on the client facing interface of the FortiGate 1000C (shown on the left) and the server facing interface on the FortiGate 3600C (shown on the right).

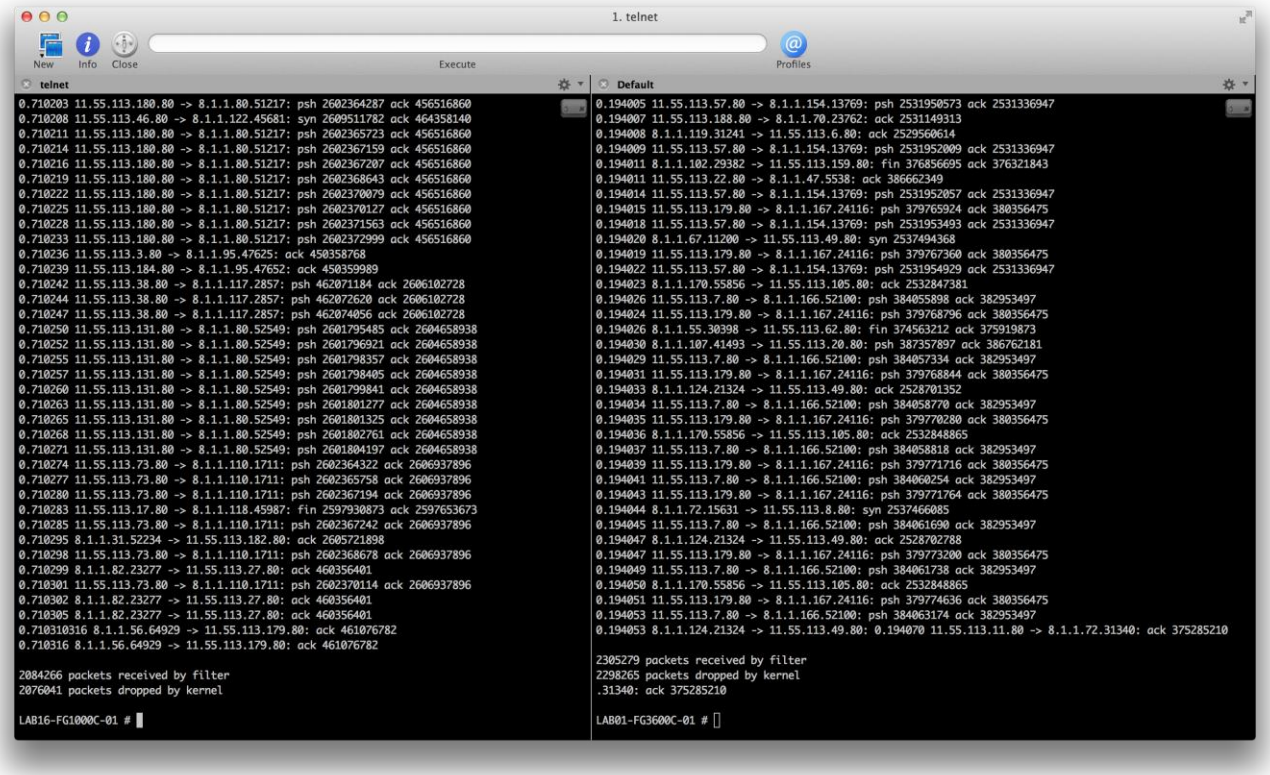


We can observe the SYN packets with some acknowledgments but we also observe “ICMP unreachable” and “need to fragment” messages. Below we can also observe that the connections are not completing since there are no FIN packets and all sessions are being reset.



Second Test

The “tcp-mss-sender” option is implemented in this test and we can observe that all sessions are completed. FIN packets show sessions closing properly and there are no resets.


1. telnet
New
Info
Close
Execute
Profiles
telnet
Default
0.710203 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602364287 ack 456516860
0.710208 11.55.113.46.80 -> 8.1.1.122.45681: syn 2609511782 ack 464358140
0.710211 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602365723 ack 456516860
0.710214 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602367159 ack 456516860
0.710216 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602367207 ack 456516860
0.710219 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602368643 ack 456516860
0.710222 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602370079 ack 456516860
0.710225 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602370127 ack 456516860
0.710228 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602371563 ack 456516860
0.710233 11.55.113.180.80 -> 8.1.1.80.51217: psh 2602372999 ack 456516860
0.710236 11.55.113.3.80 -> 8.1.1.95.47625: ack 450358768
0.710239 11.55.113.184.80 -> 8.1.1.95.47652: ack 450359989
0.710242 11.55.113.38.80 -> 8.1.1.117.2857: psh 462071184 ack 2606102728
0.710244 11.55.113.38.80 -> 8.1.1.117.2857: psh 462072620 ack 2606102728
0.710247 11.55.113.38.80 -> 8.1.1.117.2857: psh 462074056 ack 2606102728
0.710250 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601795485 ack 2604658938
0.710252 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601796921 ack 2604658938
0.710255 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601798357 ack 2604658938
0.710257 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601798405 ack 2604658938
0.710260 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601799841 ack 2604658938
0.710263 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601801277 ack 2604658938
0.710265 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601801325 ack 2604658938
0.710268 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601802761 ack 2604658938
0.710271 11.55.113.131.80 -> 8.1.1.80.52549: psh 2601804197 ack 2604658938
0.710274 11.55.113.73.80 -> 8.1.1.110.1711: psh 2602364322 ack 2606937896
0.710277 11.55.113.73.80 -> 8.1.1.110.1711: psh 2602365758 ack 2606937896
0.710280 11.55.113.73.80 -> 8.1.1.110.1711: psh 2602367194 ack 2606937896
0.710283 11.55.113.17.80 -> 8.1.1.118.45987: fin 2597930873 ack 2597653673
0.710285 11.55.113.73.80 -> 8.1.1.110.1711: psh 2602367242 ack 2606937896
0.710295 8.1.1.31.52234 -> 11.55.113.182.80: ack 2605721898
0.710298 11.55.113.73.80 -> 8.1.1.110.1711: psh 2602368678 ack 2606937896
0.710299 8.1.1.82.23277 -> 11.55.113.27.80: ack 460356401
0.710301 11.55.113.73.80 -> 8.1.1.110.1711: psh 2602370114 ack 2606937896
0.710302 8.1.1.82.23277 -> 11.55.113.27.80: ack 460356401
0.710305 8.1.1.82.23277 -> 11.55.113.27.80: ack 460356401
0.710310316 8.1.1.56.64929 -> 11.55.113.179.80: ack 461076782
0.710316 8.1.1.56.64929 -> 11.55.113.179.80: ack 461076782
2084266 packets received by filter
2076041 packets dropped by kernel
LAB16-FG1000C-01 #
0.194005 11.55.113.57.80 -> 8.1.1.154.13769: psh 2531950573 ack 2531336947
0.194007 11.55.113.188.80 -> 8.1.1.70.23762: ack 2531149313
0.194008 8.1.1.119.31241 -> 11.55.113.6.80: ack 2529560614
0.194009 11.55.113.57.80 -> 8.1.1.154.13769: psh 2531952009 ack 2531336947
0.194011 8.1.1.102.29382 -> 11.55.113.159.80: fin 376856695 ack 376321843
0.194011 11.55.113.22.80 -> 8.1.1.47.5538: ack 386662349
0.194014 11.55.113.57.80 -> 8.1.1.154.13769: psh 2531952057 ack 2531336947
0.194015 11.55.113.179.80 -> 8.1.1.167.24116: psh 379765924 ack 380356475
0.194018 11.55.113.57.80 -> 8.1.1.154.13769: psh 2531953493 ack 2531336947
0.194020 8.1.1.67.11200 -> 11.55.113.49.80: syn 2537494368
0.194019 11.55.113.179.80 -> 8.1.1.167.24116: psh 379767360 ack 380356475
0.194022 11.55.113.57.80 -> 8.1.1.154.13769: psh 2531954929 ack 2531336947
0.194023 8.1.1.170.55856 -> 11.55.113.105.80: ack 2532847381
0.194026 11.55.113.7.80 -> 8.1.1.166.52100: psh 384055898 ack 382953497
0.194024 11.55.113.179.80 -> 8.1.1.167.24116: psh 379768796 ack 380356475
0.194026 8.1.1.55.30398 -> 11.55.113.62.80: fin 374563212 ack 375919873
0.194030 8.1.1.107.41493 -> 11.55.113.20.80: psh 387357897 ack 386762181
0.194029 11.55.113.7.80 -> 8.1.1.166.52100: psh 384057334 ack 382953497
0.194031 11.55.113.179.80 -> 8.1.1.167.24116: psh 379768844 ack 380356475
0.194033 8.1.1.124.21324 -> 11.55.113.49.80: ack 2528701352
0.194034 11.55.113.7.80 -> 8.1.1.166.52100: psh 384058770 ack 382953497
0.194035 11.55.113.179.80 -> 8.1.1.167.24116: psh 379770280 ack 380356475
0.194036 8.1.1.170.55856 -> 11.55.113.105.80: ack 2532848865
0.194037 11.55.113.7.80 -> 8.1.1.166.52100: psh 384058818 ack 382953497
0.194039 11.55.113.179.80 -> 8.1.1.167.24116: psh 379771716 ack 380356475
0.194041 11.55.113.7.80 -> 8.1.1.166.52100: psh 384060254 ack 382953497
0.194043 11.55.113.179.80 -> 8.1.1.167.24116: psh 379771764 ack 380356475
0.194044 8.1.1.72.15631 -> 11.55.113.8.80: syn 2537466085
0.194045 11.55.113.7.80 -> 8.1.1.166.52100: psh 384061690 ack 382953497
0.194047 8.1.1.124.21324 -> 11.55.113.49.80: ack 2528702788
0.194047 11.55.113.179.80 -> 8.1.1.167.24116: psh 379773200 ack 380356475
0.194049 11.55.113.7.80 -> 8.1.1.166.52100: psh 384061738 ack 382953497
0.194050 8.1.1.170.55856 -> 11.55.113.105.80: ack 2532848865
0.194051 11.55.113.179.80 -> 8.1.1.167.24116: psh 379774636 ack 380356475
0.194053 11.55.113.7.80 -> 8.1.1.166.52100: psh 384063174 ack 382953497
0.194053 8.1.1.124.21324 -> 11.55.113.49.80: 0.194070 11.55.113.11.80 -> 8.1.1.72.31340: ack 375285210
2305279 packets received by filter
2298265 packets dropped by kernel
.31340: ack 375285210
LAB01-FG3600C-01 #